master 1 branch 0 tags Go to file Add file Code

SivilTaram revert the unexpected changes. 3cc800f on Nov 16, 2020 29 commits

agents	Update psquare agent saving function.	5 months ago
logs	Initialize code repo.	12 months ago
misc	Update Misc and Readme.	12 months ago
parlai	Initialize code repo.	12 months ago
scripts	Update psquare agent saving function.	5 months ago
tasks	Update selfplay script.	12 months ago
worlds	Fix a little issue.	12 months ago
.gitignore	Initialize code repo.	12 months ago
LICENSE	Update License to Microsoft. Update README	12 months ago
README.md	Update psquare agent saving function.	5 months ago
eval_f1.py	Initialize code repo.	12 months ago
eval_hits.py	Initialize code repo.	12 months ago
interactive.py	Fix the issue of interactive.py .	12 months ago
train_psquare.py	revert the unexpected changes.	5 months ago
train_receiver_model.py	Update training scripts.	12 months ago
train_transmitter.py	Update train_transmitter.py	7 months ago

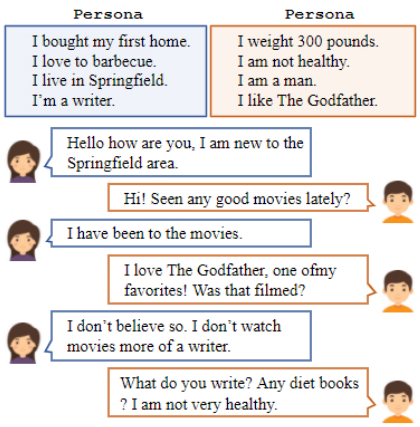
README.md

Persona-Dialogue-Generation PyTorch

State of the Art Dialogue Generation on Persona-Chat

This is the official code for our ACL 2020 paper You Impress Me: Dialogue Generation via Mutual Persona Perception.

Task & Experiments



Our target task is **Open-domain Personalized Dialogue Generation**. As shown above, two interlocutors meet for the first time and are having a conversation in order to get to know each other. The model is aware of their persona, which are explicitly described using several profile sentences, facilitating the training of chatbots with configurable and persistent personalities.

We conduct experiments on PersonaChat. And the main results on the validation dataset are as following (partial results of baselines are borrowed from here):

Setting	Model	PPL	Hits@1	F1

About

The code of ACL 2020 paper "You Impress Me: Dialogue Generation via Mutual Persona Perception"

Readme

MIT License

Releases

No releases published

Packages

No packages published

Contributors 2

SivilTaram Qian

parthpatwa

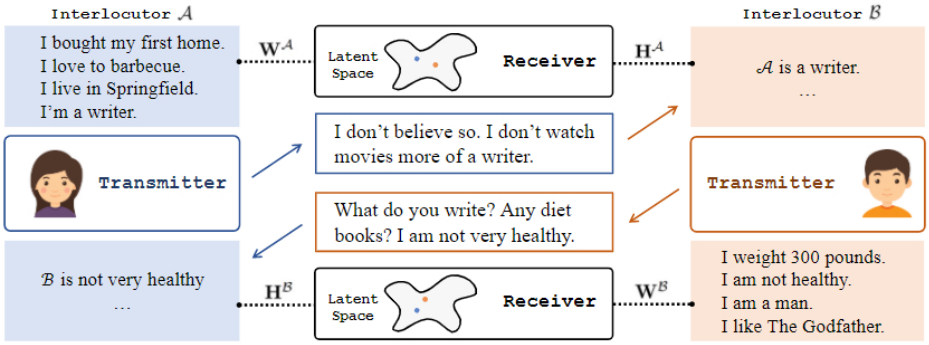
Languages

Python 86.7% HTML 7.8% JavaScript 4.8% Other 0.7%

Setting	Model	PPL	Hits@1	F1
	Ours	**15.12** 🍓	81.9	**19.77** 🍓
	Transfertransfo	17.51	**82.1** 🍓	19.09
	Lost In Conversation	-	17.3	17.79
	Seq2seq-Attention	35.07	12.5	16.82
Original	Language Model	50.67	-	16.30
	Generative Profile Memory	35.01	10.2	16.29
	Dually Interactive Matching	-	78.8	-
	KV Profile Memory	-	54.8	14.25

Details about each baseline are shown in our paper.

Model Quick Overview



In this paper, we propose a a transmitter-receiver based framework with the aim of explicitly modelling **Persona understanding**, in other words, **Mutual Persona Perception**.

It is based on the following motivation: the two interlocutors foster understanding either by raising persona-related topics, `Seen any good movies lately?`, or by revealing their own personas through answering questions, `I don't watch movies more of a writer.`. The efforts to build understanding keep the conversation flowing.

Install Dependencies

Python Environment

First of all, you should setup a python environment. This code base has been tested under python 3.x, and we officially support python 3.7.

After installing python 3.7, we strongly recommend you to use `virtualenv` (a tool to create isolated Python environments) to manage the python environment. You could use following commands to create a environment.

```
python -m pip install virtualenv
virtualenv venv
```

Activate Virtual Environment

Then you should activate the environment to install the dependencies. You could achieve it via using the command as below. (Please change $ENV_FOLDER to your own virtualenv folder path, e.g. venv)

```
$ENV_FOLDER\Scripts\activate.bat (Windows)
source $ENV_FOLDER/bin/activate (Linux)
```

Install PyTorch

The most important requirements of our code base are `pytorch == 1.0.1` and `tensorboardX`. You should install them at first.

Install Custom Dependencies

Besides pytorch, our code is mainly based on ParlAI and Huggingface's transformers (pytorch-pretrained-bert v0.6.2) library. As they are under active development, for the purpose to reproduce our results, we provide two custom repos to install them. It is worth noting that we also modify a little on Huggingface's code to achieve the auxiliary task `Next Utterance Prediction` (See Section 3.1 in our paper), and more details on changes could be seen here. Assuming you current working directory is `./`, you can run the following script to install them:

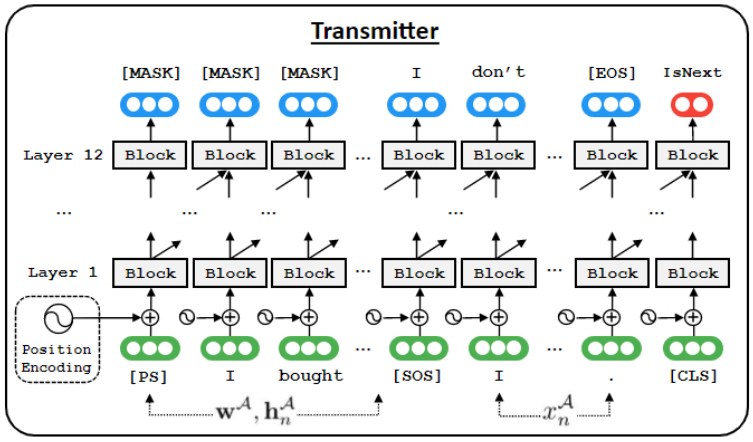
```
cd ..
git clone https://github.com/SivilTaram/transformers.git
```

```
cd transformers
python setup.py install
cd ..
git clone https://github.com/SivilTaram/ParlAI.git
cd ParlAI
python setup.py install
cd ..
cd Persona-Dialogue-Generation
```

Training

We provide three files to train `Transmitter`, `Receiver` and `PSquare` (details can be found in our paper). And the corresponding training scripts and commands are as below.

Training Transmitter



The transmitter is based OpenAI's GPT model. The default hyper-parameters are expected to reproduce our paper results (if not, please open an issue or contact me via email). Therefore, you could use the following command to train a transmitter. The script will automatically download the PersonaChat dataset into the `./data/ConvAI2` folder. Note that we use the `train_self_(original|revised).txt` to train Transmitter.

```
python train_transmitter.py
```

If you want to train our model on your own collected data, please follow the format of PersonaChat to update the file `train_self_(original|revised).txt`. One example is as following:

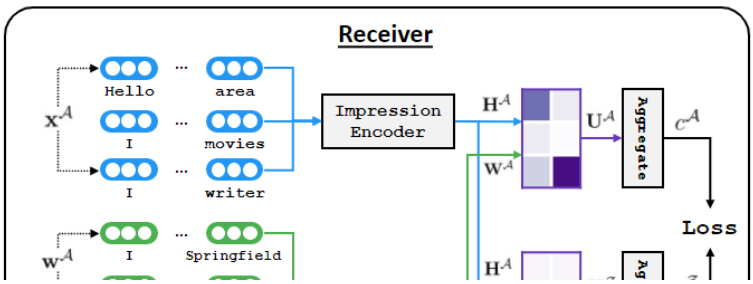
```
1 your persona: i like to remodel homes.
2 your persona: i like to go hunting.
3 your persona: i like to shoot a bow.
4 your persona: my favorite holiday is halloween.
5 hi , how are you doing ? i'm getting ready to do some cheetah chasing to stay in shape .      you must be ve
6 i am ! for my hobby i like to do canning or some whittling .  i also remodel homes when i am not out bow hun
7 that's neat . when i was in high school i placed 6th in 100m dash !   that's awesome . do you have a favorit
8 i do not . but i do have a favorite meat since that is all i eat exclusively .        what is your favorite
9 i would have to say its prime rib . do you have any favorite foods ?  i like chicken or macaroni and cheese
10 do you have anything planned for today ? i think i am going to do some canning .     i am going to watch fo
11 i think i will can some jam . do you also play footfall for fun ?    if i have time outside of hunting and
```

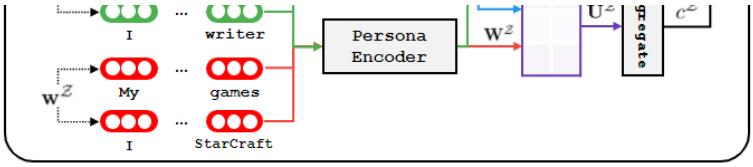
Training Receiver

If you have downloaded the ConvAI2 dataset, you could use `./tasks/convai2receiver/build_data.py` to build the dataset for receiver:

```
python tasks/convai2receiver/build_data.py
```

> You may receive the hint `Tried to build dictionary but --dict-file is not set. Set this param so the dictionary can be saved.` after running the above script. But don't worry about it since it is an expected warnning.





The backbone of our Receiver is BERT. And it is trained via a weak-supervision fashion. You could train the Receiver model as:

```
python train_receiver.py
```

Training PSquare

At first you should prepare the self-play datset using the following command:

```
python tasks/convai2/build_data.py
```

Before training PSquare, you should have a trained transmitter and receiver. Specifying the model names in line 33-42 in `train_psquare.py`, you can run the following script to execute the self-play procedure.

```
python train_psquare.py
```

Note that we use two cards to train our PSquare bot to speed up. If you do not have two or more GPU cards, you could comment lines 444-445 in `agents/psquare/psquare.py`.

```
self.coherent_model.cuda("cuda:1")
self.language_model.cuda('cuda:1')
```

Trained Model Weights

We also provide trained PSquare weights for reproducing our experimental results in the paper.

- Trained model weights under the Original setting:
  https://www.dropbox.com/s/ozw9xmfv4f0tud9/psqaure_original.zip?dl=0
- Trained model weights under the Revised setting:
  https://www.dropbox.com/s/bbvamaj9r019wsw/psqaure_revised.zip?dl=0

Please create a directory `./tmp/psquare`, and unzip the model zipped files into the directory as:

```
| -- tmp
    | -- psquare
        | -- psqaure_original.model
        | -- psqaure_original.model.opt
        | -- psqaure_original.model.best_valid
```

Then you could directly evaluate it using the following evaluation scripts.

Evaluation

You could run `eval_f1.py`, `eval_hits.py` to obtain the `F1`, `Hits@1` for either Transmitter or PSquare. The evaluation logs on our provided model weights can be found in the folder `./logs/`.

As for the `ppl` metric, you could run the training script on a trained model file to fake the continuation of training. The restoring will first validate and report `ppl` on the validation dataset.

Acknowledgement

We will first thank for the PersonaChat for publishing such a great dataset.

The `parlai` module is modified from ParlAI. Thanks them for their huge contributions on developing such a great conversational platform (*Attention: usage on this module follows its open source License*) ! Also many thanks for Huggingface's transformer library!

Contact

You could reach me via my email: qian dot liu at buaa dot edu dot cn. Or just feel free to open an issue :)

Citation

Please consider citing our paper if it is helpful to you :)

```
@inproceedings{liu-etal-2020-personachat,
```

```
    title = "You Impress Me: Dialogue Generation via Mutual Persona Perception",
    author = "Liu, Qian  and
      Chen, Yihong  and
      Chen, Bei  and
      Lou, Jian-Guang  and
      Chen, Zixuan  and
      Zhou, Bin  and
      Zhang, Dongmei",
    booktitle = "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics",
    month = july,
    year = "2020",
    publisher = "Association for Computational Linguistics"
}
```